

For answer script - EVERY answer sheet MUST include (a) full student name, (b) student number, (c) subject code, (d) subject group and (e) page number.

Name:	<u>Chung Lok Ming</u>	Student No.:	<u>20008848A</u>
Subject Code	<u>SEHH2239</u>	Subject Lecture Group:	<u></u>
Page No.:	<u>1</u>	Total no. of pages:	<u>13</u>

Declaration of Original Work

By submitting the answer script of this assignment to the subject lecturer through Moodle Centralized Group, you hereby declare that the work in the answer sheet is completely your own work. No part of the answer sheet is taken from other people's work without giving them credit. All references have been clearly cited.

You understand that an infringement of this declaration leaves you subject to disciplinary actions such as mark deduction, disqualification or even expulsion by the College.

If necessary, students may be invited to provide more information on their submission.

(Please refer to the relevant section(s) on plagiarism of the Student Handbook.)

Instructions to Students:

1. Please refer to assignment specification for the submission method
2. Show all your work clearly and neatly. Marks will be deducted for untidy work.

Answer ALL questions.

For answer script - EVERY answer sheet MUST include (a) full student name, (b) student number, (c) subject code, (d) subject group and (e) page number.

Name: Chung Lok Ming

Student No.: 20008848A

Subject Code SEHH2239

Subject Lecture Group:

Page No.: 2

Total no. of pages: 13

Answer for Question 1

A= 08 b=88 c=84 d=48

a) + * a b * c d -> +*8 88 * 84 48

b) a*b + c*d -> 8*88 + 84*48

c) 4736

For answer script - EVERY answer sheet MUST include (a) full student name, (b) student number, (c) subject code, (d) subject group and (e) page number.

Name: Chung Lok Ming

Student No.: 20008848A

Subject Code: SEHH2239

Subject Lecture Group:

Page No.: 3

Total no. of pages: 13

Answer for Question 2

a)

```
#q2a
class MyMailQue:
    def __init__(self) :
        self.mails = []
        self.size = 0

    def enqueue(self, data):
        self.mails.append(data)
        self.size += 1

    def dequeue(self):
        if self.size > 1:
            popped = self.mails[0]
            self.mails = self.mails[1:]
            self.size -= 1
            return popped
        elif self.size == 1:
            popped = self.mails[0]
            self.mails = []
            self.size -= 1
            return popped
        else:
            return None

    def display(self):
        for i in self.mails:
            print(i)
```

For answer script - EVERY answer sheet MUST include (a) full student name, (b) student number, (c) subject code, (d) subject group and (e) page number.

Name: Chung Lok Ming

Student No.: 20008848A

Subject Code: SEHH2239

Subject Lecture Group:

Page No.: 4

Total no. of pages: 13

b)

```
def searchAllPath(self, root):
    if root is None:
        return

    self.store.append(root)
    if root.left_node or root.right_node:
        self.searchAllPath(root.left_node)
        self.searchAllPath(root.right_node)
    else:
        weight = 0
        for i in self.store:
            if i.data == "end":
                print(f'{i.data}', end='')
                print(f'\t{weight}')
            else:
                print(f"{i.data}->", end='')
                if i.data != 'start':
                    weight += i.data

        self.store.pop()
```

For answer script - EVERY answer sheet MUST include (a) full student name, (b) student number, (c) subject code, (d) subject group and (e) page number.

Name:	Chung Lok Ming	Student No.:	20008848A
Subject Code	SEHH2239	Subject Lecture Group:	
Page No.:	5	Total no. of pages:	13

c)

```
57 Mailserver = MailBox()
58 Mailserver.createMailBox("mymail")
59 Mailserver.createMailBox("classmateMail")
60
61 Mailserver.send("mymail", "classmateMail", "Hello ! How are you ?" )
62 Mailserver.send("mymail", "classmateMail", "Please prepare assignment for Data structure ")
63 Mailserver.send("classmateMail", "mymail", "I got your assignment and will prepare it ! ")
64 Mailserver.send("mymail", "classmateMail", "Thanks you and see you later ! ")
65
66 mail = Mailserver.receive("mymail")
67 print("My receive mail :" + str(mail) )
68
69 mail = Mailserver.receive("classmateMail")
70 print("My classmate receive mail :" + str(mail) )
71
72
73 #test case 1
74 Test1Server = MailBox()
75 Test1Server.createMailBox('test1')
76 Test1Server.createMailBox('test2')
77
78 Test1Server.send('test1', 'test2', "Email 1 for testing")
79
80 mail = Test1Server.receive('test2')
81 print("test2 receive mail :" + str(mail) )
82
83 #test case 2
84 Test2Server = MailBox()
85 Test2Server.createMailBox('test3')
86 mail = Test2Server.receive('test3')
87 print("test3 receive mail :" + str(mail) )
88
89 #test case 3
90 Test3Server = MailBox()
91 mail = Test3Server.receive('test4')
92 print("test4 receive mail :" + str(mail) )
```

For answer script - EVERY answer sheet MUST include (a) full student name, (b) student number, (c) subject code, (d) subject group and (e) page number.

Name:	Chung Lok Ming	Student No.:	20008848A
Subject Code	SEHH2239	Subject Lecture Group:	
Page No.:	6	Total no. of pages:	13

Answer for Question 3

8 88 84 48 80 840 880

a=8

b=88

c=84

d=48

e=80

f=840

g=880

a)

i)

```
class Node:
    def __init__(self, data):
        self.data = data
        self.left_node = None
        self.right_node = None
```

ii)

```
106     def searchPath(self, root):
107         self.store.append(root)
108         if root.data == 'end':
109             for item in self.store:
110                 if item.data == "end":
111                     print(f"{item.data}")
112                 else:
113                     print(f"{item.data}->", end="")
114             return
115         else:
116             if root.left_node:
117                 self.searchPath(root.left_node)
118             elif root.right_node:
119                 self.searchPath(root.right_node)
```

b)

For answer script - EVERY answer sheet MUST include (a) full student name, (b) student number, (c) subject code, (d) subject group and (e) page number.

Name: Chung Lok Ming

Student No.: 20008848A

Subject Code: SEHH2239

Subject Lecture Group:

Page No.: 7

Total no. of pages: 13

```
121     def searchAllPath(self, root):
122         if root is None:
123             return
124
125         self.store.append(root)
126
127         if root.left_node is None and root.right_node is None:
128             weight = 0
129             for i in self.store:
130                 if i.data == "end":
131                     print(f'{i.data}', end='')
132                     print(f'\t{weight}')
133                 else:
134                     print(f"{i.data}->", end='')
135                     if i.data != 'start':
136                         weight += i.data
137
138         self.searchAllPath(root.left_node)
139         self.searchAllPath(root.right_node)
140
141         self.store.pop()
```

For answer script - EVERY answer sheet MUST include (a) full student name, (b) student number, (c) subject code, (d) subject group and (e) page number.

Name:	Chung Lok Ming	Student No.:	20008848A
Subject Code	SEHH2239	Subject Lecture Group:	
Page No.:	8	Total no. of pages:	13

c)

```
143     def searchPathByWeight(self, root, end):
144         if root is None:
145             return
146
147         self.store.append(root)
148         if root.left_node is None and root.right_node is None:
149             weight = 0
150             for i in self.store:
151                 if i.data == "end":
152                     print(f'{i.data}', end='')
153                     print(f'\t{weight}')
154                 else:
155                     print(f"{i.data}->", end='')
156                     if i.data != 'start':
157                         weight += i.data
158
159             if root.left_node and root.right_node:
160                 if root.left_node.data < root.right_node.data:
161                     self.searchPathByWeight(root.left_node, end)
162                 else:
163                     self.searchPathByWeight(root.right_node, end)
164 +
165         self.store.pop()
```

d)

advantage of searchAllPath must be able to choose a shortest path by listing all the path. And for the advantage of searchpathbyweight time complexity is lower. And for the disadvantage of searchAllPath is that the time complexity will need a longer time to run all the path. As for the disadvantages searchpathbyweight may not find the shortest path since it is dependent on the next loop and won't take another path.

For answer script - EVERY answer sheet MUST include (a) full student name, (b) student number, (c) subject code, (d) subject group and (e) page number.

Name: Chung Lok Ming

Student No.: 20008848A

Subject Code: SEHH2239

Subject Lecture Group:

Page No.: 9

Total no. of pages: 13

```

166 #test 1
167 Start_node = Node('start')
168
169 a=8
170 b=88
171 c=84
172 d=48
173 e=80
174 f=840
175 g=880
176
177 Start_node = Node('start')
178 b_node = Node(b)
179 d_node = Node(d)
180 e_node = Node(e)
181 End_node = Node('end')
182
183 Start_node.left_node = b_node
184 b_node.left_node = d_node
185 b_node.right_node = e_node
186 d_node.right_node = End_node
187 e_node.left_node = End_node
188
189
190 mypath = Path()
191 mypath.searchPath(Start_node)
192
193 print('mypath.searchAllPath: ')
194 mypath.store = []
195 mypath.searchAllPath(Start_node)
196
197 print('mypath.searchPathByWeight')
198 mypath.store = []
199 mypath.searchPathByWeight(Start_node, End_node)
200
201
202 start = Node('start')
203 a_node = Node(a)
204 b_node = Node(b)
205 b2 = Node(b)
206 c_node = Node(c)
207 c2 = Node(c)
208 d_node = Node(d)
209 d2 = Node(d)
210 e_node = Node(e)
211 e2 = Node(e)
212 f_node = Node(f)
213 f2 = Node(f)
214 g_node = Node(g)
215 end = Node('end')
216
217 start.left_node = b_node
218 start.right_node = c_node
219
220 b_node.left_node = d_node
221 b_node.right_node = e2
222
223 c_node.left_node = f2
224 c_node.right_node = g_node
225
226 d_node.left_node = f_node
227 d_node.right_node = e_node
228
229 e2.right_node = d2
230
231 f2.right_node = c2
232
233 g_node.left_node = b2
234 g_node.right_node = a_node
235
236 f_node.right_node = end
237 e_node.right_node = end
238 d2.right_node = end
239 c2.left_node = end
240 b2.left_node = end
241 a_node.left_node = end
242
243 mypath2 = Path()
244 print('mypath2')
245 mypath2.searchAllPath(start)
246
247 mypath2.store = []
248 print('mypath2')
249 mypath2.searchPathByWeight(start, end)
250
251 nodeStart = Node('start')
252
253 node6915 = Node(6915)
254 node300 = Node(300)
255 node59 = Node(59)
256 node31 = Node(31)
257 node21 = Node(21)
258 node12 = Node(12)
259 node45 = Node(45)
260 node786 = Node(786)
261 node86 = Node(86)
262 node68 = Node(68)
263 node88 = Node(88)
264 node48 = Node(48)
265 node689 = Node(689)
266 node777 = Node(777)
267 node1 = Node(1)
268 node996 = Node(996)
269 node8964 = Node(8964)
270 node101 = Node(101)
271 node911 = Node(911)
272 node899 = Node(899)
273 node517 = Node(517)
274 node7 = Node(7)
275 node167 = Node(167)
276 node169 = Node(169)
277 node612 = Node(612)
278 node831 = Node(831)
279 node721 = Node(721)
280 node609 = Node(609)
281 node107 = Node(107)
282 node1911 = Node(1911)
283 node1001 = Node(1001)
284
285 nodeEnd = Node('end')
286
287 nodeStart.left_node = node6915
288 nodeStart.right_node = node300
289
290 node6915.left_node = node59
291 node6915.right_node = node31
292
293 node59.left_node = node45
294 node59.right_node = node786
295
296 node45.left_node = node689
297 node45.right_node = node777

```

For answer script - EVERY answer sheet MUST include (a) full student name, (b) student number, (c) subject code, (d) subject group and (e) page number.

Name:	Chung Lok Ming	Student No.:	20008848A
Subject Code	SEHH2239	Subject Lecture Group:	
Page No.:	10	Total no. of pages:	13

```

299 node689.left_node = node101
300 node689.right_node = node911
301
302 node31.left_node = node86
303
304 node86.left_node = node1
305
306 node1.left_node = node899
307 node1.right_node = node517
308
309 node899.left_node = node612
310 node899.right_node = node831
311
312 node612.left_node = node107
313
314 node107.left_node = node1911
315 node107.right_node = node1001
316
317 node517.left_node = node721
318 node517.right_node = node609
319
320 node300.left_node = node21
321 node300.right_node = node12
322
323 node21.left_node = node68
324 node21.right_node = node88
325
326 node68.right_node = node996
327
328 node996.right_node = node7
329
330 node12.right_node = node48
331
332 node48.left_node = node8964
333
334 node8964.left_node = node167
335 node8964.right_node = node169
336
337 node101.left_node = nodeEnd
338 node911.left_node = nodeEnd
339 node1911.left_node = nodeEnd
340 node1001.left_node = nodeEnd
341 node831.left_node = nodeEnd
342 node721.left_node = nodeEnd
343 node609.left_node = nodeEnd
344 node7.left_node = nodeEnd
345 node167.left_node = nodeEnd
346 node169.left_node = nodeEnd
347 node88.left_node = nodeEnd
348 node777.left_node = nodeEnd
349 node786.left_node = nodeEnd
350
351 mypath3 = Path()
352 print('mypath3')
353 mypath3.searchAllPath(nodeStart)
354
355 mypath3.store = []
356 print('mypath3')
357 mypath3.searchPathByWeight(nodeStart, nodeEnd)
358
359 #test 2
360 node_start = Node('start')
361 node_end = Node('end')
362
363 node_1 = Node(1)
364 node_2 = Node(2)
365 node_3 = Node(3)
366 node_4 = Node(4)
367 node_5 = Node(5)
368 node_6 = Node(6)
369 node_7 = Node(7)
370 node_8 = Node(8)
371 node_9 = Node(9)
372 node_10 = Node(10)
373 node_11 = Node(11)
374 node_12 = Node(12)
375 node_13 = Node(13)
376 node_14 = Node(14)
377 node_15 = Node(15)
378 node_16 = Node(16)
379 node_17 = Node(17)
380 node_18 = Node(18)
381 node_19 = Node(19)
382 node_20 = Node(20)
383 node_21 = Node(21)
384 node_22 = Node(22)
385 node_23 = Node(23)
386 node_24 = Node(24)
387 node_25 = Node(25)
388 node_26 = Node(26)
389 node_27 = Node(27)
390 node_28 = Node(28)
391 node_29 = Node(29)
392 node_30 = Node(30)
393 node_31 = Node(31)
394
395 node_start.left_node = node_1
396 node_start.right_node = node_2
397
398 node_1.left_node = node_3
399 node_1.right_node = node_4
400
401 node_3.left_node = node_7
402 node_3.right_node = node_8
403
404 node_7.left_node = node_15
405 node_7.right_node = node_16
406
407 node_8.left_node = node_17
408 node_8.right_node = node_18
409
410 node_4.left_node = node_9
411 node_4.right_node = node_10
412
413 node_9.left_node = node_19
414 node_9.right_node = node_20
415
416 node_10.right_node = node_21
417
418 node_2.left_node = node_5
419 node_2.right_node = node_6
420
421 node_5.left_node = node_11
422 node_5.right_node = node_12
423
424 node_11.left_node = node_22

```

For answer script - EVERY answer sheet MUST include (a) full student name, (b) student number, (c) subject code, (d) subject group and (e) page number.

Name: Chung Lok Ming

Student No.: 20008848A

Subject Code: SEHH2239

Subject Lecture Group:

Page No.: 11

Total no. of pages: 13

```
426     node_12.right_node = node_23
427
428     node_6.left_node = node_13
429     node_6.right_node = node_14
430
431     node_13.left_node = node_24
432
433     node_14.left_node = node_25
434     node_14.right_node = node_26
435
436     node_26.left_node = node_27
437     node_26.right_node = node_28
438
439     node_27.left_node = node_29
440     node_27.right_node = node_30
441
442     node_28.right_node = node_31
443
444     node_15.right_node = node_end
445     node_16.right_node = node_end
446     node_17.right_node = node_end
447     node_18.right_node = node_end
448     node_19.right_node = node_end
449     node_20.right_node = node_end
450     node_21.right_node = node_end
451     node_22.left_node = node_end
452     node_23.left_node = node_end
453     node_24.left_node = node_end
454     node_25.left_node = node_end
455     node_29.left_node = node_end
456     node_30.left_node = node_end
457     node_31.left_node = node_end
458
459     mypath4 = Path()
460     print('mypath4')
461     mypath4.searchAllPath(node_start)
462
463     mypath4.store = []
464     print('mypath4')
465     mypath4.searchPathByWeight(node_start, node_end)
466
```

For answer script - EVERY answer sheet MUST include (a) full student name, (b) student number, (c) subject code, (d) subject group and (e) page number.

Name: Chung Lok Ming

Student No.: 20008848A

Subject Code: SEHH2239

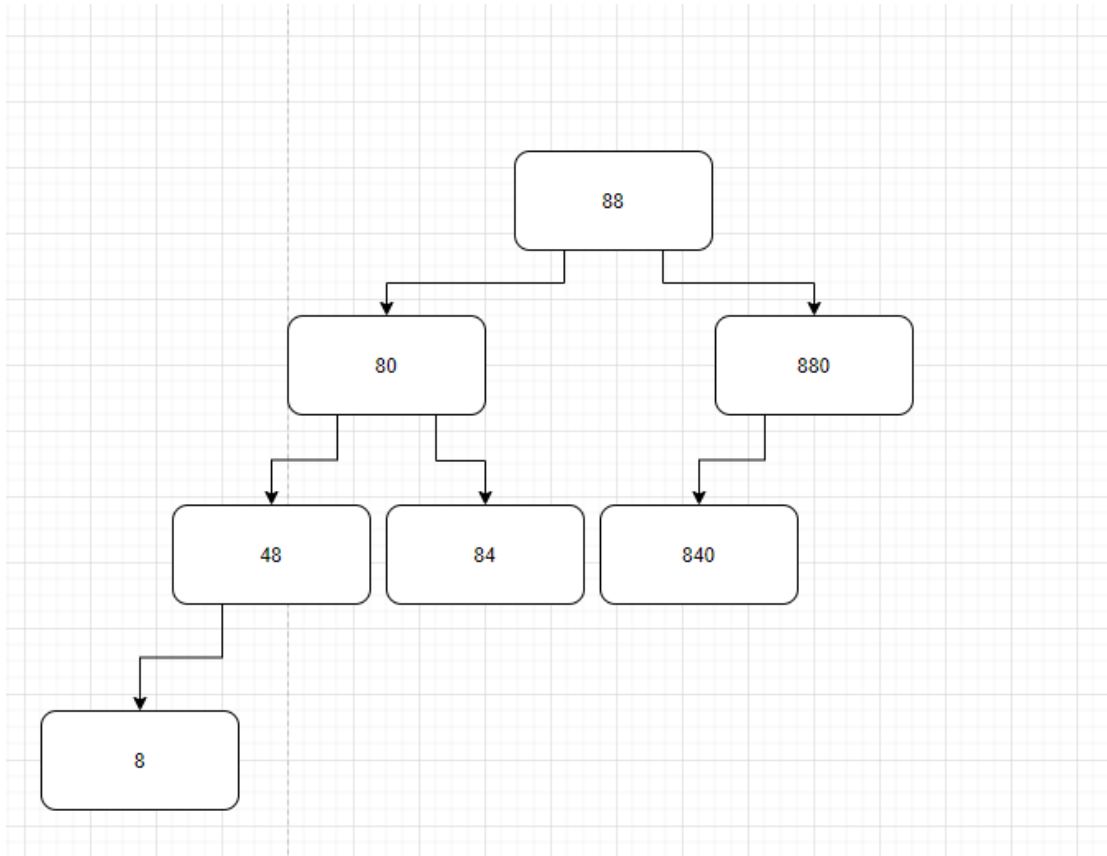
Subject Lecture Group:

Page No.: 12

Total no. of pages: 13

Answer for Question 4

A)



For answer script - EVERY answer sheet MUST include (a) full student name, (b) student number, (c) subject code, (d) subject group and (e) page number.

Name: Chung Lok Ming

Student No.: 20008848A

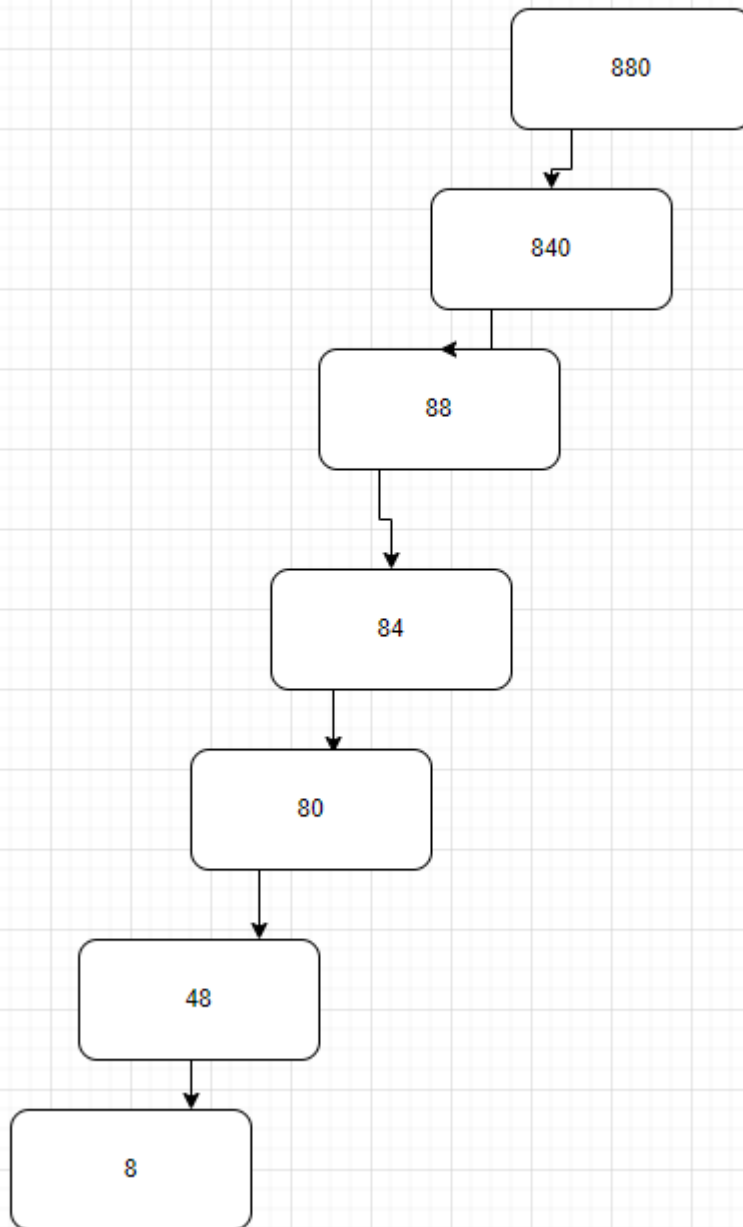
Subject Code: SEHH2239

Subject Lecture Group:

Page No.: 13

Total no. of pages: 13

b)



C)88, 80, 880, 48, 84, 840, 8